

# Shapes Subclasses Assignment

In this assignment you develop a program that that implements inheritance, pure virtual member functions, abstract base classes and base class pointers.

## BasicShapeClass

Design an abstract base class named BasicShapeClass that has the following:

- Private member variable
  - double area
- Public member functions
  - //accessors and mutators
    - getArea()
    - setArea()
- Pure virtual member function
  - calculateArea()
  - display()

## CircleShapeSubClass

Design a class named CircleShapeSubClass that is derived from BasicShapeClass that has the following:

- Private members:
  - Const double PI
  - double radius
- Public members
  - Constructor
    - CircleShapeSubClass(double radius)
  - Pure Virtual Function Implementations
    - double calculateArea()
      - Returns pi (3.14159) x radius x radius
  - void display()

## RectangleShapeSubClass

Design a class named RectangleShapeSubClass that is derived from BasicShapeClass that has the following:

- Private member variable
  - double width
  - double length
- Public member functions
  - Constructor
    - RectangleShapeSubClass(double width, double length)
  - Pure Virtual Function Implementations
    - Double calculateArea()
      - Returns width x length
  - void display()

# Shapes Subclasses Assignment

## TriangleShapeSubClass

Design a class named TriangleShapeSubClass that is derived from BasicShapeClass that has the following:

- Private member variable
  - double height
  - double base
- Public member functions
  - Constructor
    - TriangleShapeSubClass(height, base)
  - Pure Virtual Function Implementations
    - Double calculateArea()
      - Returns  $\frac{1}{2}$  base x height
  - void display()

The following files must be used in the assignment:

### .h Files

All .h include files must have appropriate ifndef, define, endif guards.

All .h include files must have using namespace std.

All .h include files must contains any necessary include files that are needed

Include Files:

- BasicShapeClass.h
  - Contains the BasicShapeClass
  - Inline function definitions
    - All member functions for all member functions in BasicShapeClass
- CircleShapeSubClass.h
  - Contains the includes "BasicShapes.h", "LineInfo.h".
  - Contains the CircleShapeSubClass
  - Inline function definitions
    - All member functions for all member functions in CircleShapeSubClass
- RectangleShapeSubClass.h
  - Contains the includes "BasicShapes.h", "LineInfo.h".
  - Contains the RectangleShapeSubClass
  - Inline function definitions
    - All member functions for all member functions in RectangleShapeSubClass
- TriangleShapeSubClass.h
  - Contains the includes "BasicShapes.h", "LineInfo.h".
  - Contains the TriangleShapeSubClass
  - Inline function definitions
    - All member functions for all member functions in TriangleShapeSubClass
- LineInfo.h
  - Used to create strings for standard exception handing
  - #include <stdexcept>
  - #include <iostream>
- main.cpp
  - Contains main function
  - Include all required include .h files used in the assignment

# Shapes Subclasses Assignment

## Validation using Exception Handling

Exception handling will use the c++ standard exception handler to handle exceptions for shape object attributes having negative values.

You must use the provided LineInfo.h file to handle exceptions.

Exceptions with try catch blocks only exist in the main program to validate input.

Below is an example of using domain\_error with LineInfo in the library

```
try {  
:  
    if (radius < 0)  
        throw domain_error(LineInfo("Radius less than zero", __FILE__, __LINE__));  
:  
}//try
```

You must use a catch handler after a try block that uses e.what to describe the string used in the throws that are generated by domain\_error().

```
catch (exception& e) {  
    cout << e.what() << endl;  
    cout << endl << "Press the enter key once or twice to leave..." << endl;  
    cin.ignore(); cin.get();  
    exit(EXIT_FAILURE);  
} //catch
```

# Shapes Subclasses Assignment

## main() Outline

Include BasicShapeClass.h, CircleShapeSubClass.h, RectangleShapeSubClass.h, TriangleShapeSubClass.h, and LineInfo.h.

Set using namespace std.

Declare a shape count variable.

Declare a constant integer that has the number of shapes.

Declare a dynamically allocated double pointer shape array of the Basic Shape Class using the new operator.

Display a simple message that states to have the user choose from three shapes.

Design a loop:

    Display a simple menu to the user to enter the type of shape

    Check for a valid menu choice, if its invalid display an error message and try the menu again.

    Parse the menu choice:

    Get the shape attributes for the shape type chosen

    Check the user inputted shape attributes using domain error

    If there are not any input attribute errors,

        dynamically assign into shape Array a new dynamic shapeobject pointer for each of the three shape types chosen.

until three valid shape types entries are done

Use the virtual display() function on the object pointers in the shapeArray to dynamically bind using display() the attributes for each of the type object in the shape array

End the try block

Design the catch block as shown in this assignment.

## Implementation Tips

In the derived class constructors, you should call calculateArea and then call setArea to initialize the area for each derived class.

You will implement a shape array double array of pointers to Basic Shape Class.

A sample run is provided at the end of this assignment.

Use the entered values to check is your program running correctly.

# Shapes Subclasses Assignment

Choose 3 shapes and enter attributes,  
when all 3 shapes are entered the calculated results for all 3 shapes will display

Choose your shape :

1. Circle
2. Rectangle
3. Triangle

Enter your choice : 4

Wrong menu value : 4

Please try again....

Choose your shape :

1. Circle
2. Rectangle
3. Triangle

Enter your choice : 1

Enter the circles radius : 3

Choose your shape :

1. Circle
2. Rectangle
3. Triangle

Enter your choice : 2

Enter the rectangles width, length : 3 4

Choose your shape :

1. Circle
2. Rectangle
3. Triangle

Enter your choice : 3

Enter the triangles height, base : 5 4

Shape is a Circle :

Radius : 3

Area : 28.2743

Shape is a Rectangle :

Width : 3

Length : 4

Area : 12

# Shapes Subclasses Assignment

Shape is a Triangle:

Height : 5

Base : 4

Area : 10